Module 1: Nonparametric Preliminaries

Selecting Smoothing Parameters

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 8th, 2014

Smoothing Parameter

- In both ridge and lasso regression, we saw that the parameter λ controlled the solution
  - Often, can straightforwardly equate with effective degrees of freedom
- Which λ (→ estimator) should we choose???

Want good predictions

↑ Linear smoothers

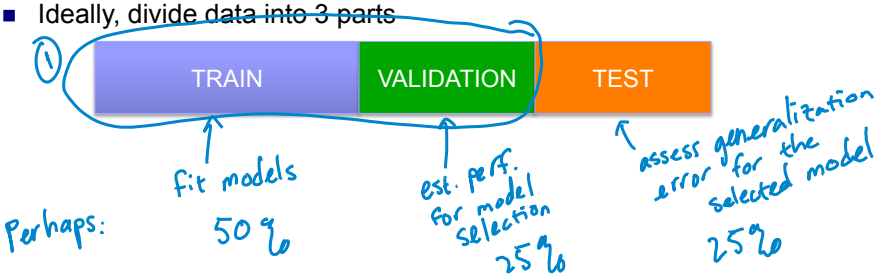
$\nu_\lambda = tr(L_\lambda)$

↑ "hat matrix"

Two Goals

1. **Model Selection:** estimating the performance of models in order to select the best one
   - E.g., choosing λ
2. **Model Assessment:** having chosen a final model, estimate its prediction error (generalization error) on new data

- Ideally, divide data into 3 parts



Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

- Recall metrics for assessing the performance of an estimator…

Measuring Predictive Performance

- Assume estimate $\hat{f}_n(\cdot)$ based on training data $y_1, \ldots, y_n$
- The ***generalization error*** provides a measure of predictive performance

$$GE(\hat{f}_n) = E_{Y,X}\left[L(Y, \hat{f}_n(X))\right]$$

Measuring Predictive Performance

- Assume L_2 loss
- Averaging over repeat training sets $\mathbf{Y_n} = Y_1, \ldots, Y_n$ we get the ***predictive risk*** at x^*

$$E_{Y^*,\mathbf{Y}_n}\left[(Y^* - \hat{f}_n(x^*))^2\right] = E_{Y^*,Y_n}\left[(Y^* - f(x^*) + f(x^*) - \hat{f}_n(x^*))^2\right]$$

$$= E_{Y^*}\left[(Y^* - f(x^*))^2\right] + E_{Y_n}\left[(\hat{f}_n(x^*) - f(x^*))^2\right] + 2E_{Y^*}\left[Y^* - f(x^*)\right] E_{Y_n}\left[\hat{f}_n(x) - f(x^*)\right]$$

$$= \sigma^2 + MSE(\hat{f}_n(x^*))$$

"irreducible error" "risk"

- Recall $MSE[\hat{f}_n(x)] = \text{bias}(\hat{f}_n(x))^2 + \text{var}(\hat{f}_n(x))$

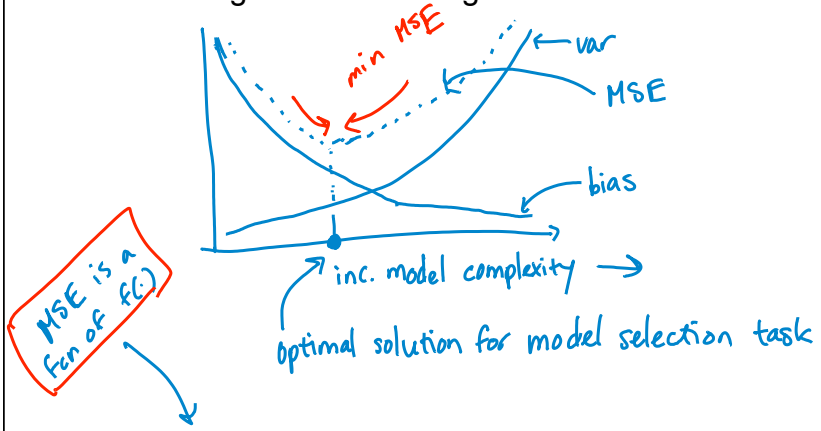
Measuring Predictive Performance

- Finally, let's average over covariates *x* (focus on MSE bc can't avoid σ^2)
  - ***Integrated MSE*** $\int MSE(\hat{f}_n(x))\,p(x)\,dx$ — summary over all inputs
  - ***Average MSE*** $\frac{1}{n}\sum_{i=1}^{n} MSE(\hat{f}_n(x_i))$ — Monte Carlo est: $X_i \sim P$ $i = 1, \ldots, n$
- Note: ***avg. pred. risk =*** σ^2 + ***avg. MSE*** $\left[\frac{1}{n}\sum_{i=1}^{n} E_{Y_i^*, Y_n}\left[(Y_i^* - \hat{f}_n(x_i))^2\right]\right]$ ← still our focus
  - Y_i^*: new obs. at X_i; Y_n: training data

Bias-Variance Tradeoff

recall polynomial reg. example

- Minimizing risk = balancing bias and variance



- Note: *f(x) is unknown, so cannot actually compute MSE*

Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

index λ

- We saw that minimizing (average) prediction error can be equated with minimizing (average) MSE

est. from training data of size n using λ

- With a validation set, we can estimate the prediction error

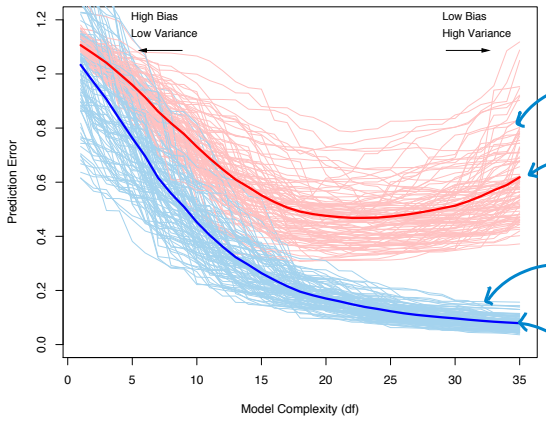
$$\frac{1}{m}\sum_{i=1}^{m}\left(y_i^* - \hat{f}_n^{\lambda}(x_i^*)\right)^2$$

↑ Obs. in validation set of size m

In Practice…

- Minimizing risk = balancing bias and variance

"test" data here is our validation set



error est. on test data

avg.

error estimated on training data

avg. over training sets

From Hastie, Tibshirani, Friedman

Data Scarce Approximations

- Often, we do not have enough data to form suitably sized training and validation sets
  - What is a good training/test split? Sensitivity?
  - Typically want to use as much data for training as possible
- Rely on other approximations using in-sample data

Approx 1: Training Data Only

DON'T USE THIS

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Use training error

$$\min \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_n^{\lambda}(x_i))^2$$

training error = $\frac{RSS}{n}$

↑ training obs.

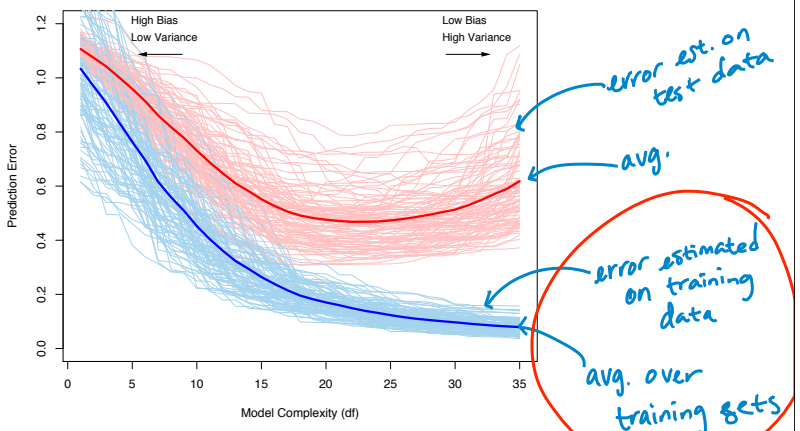
BAD biased downwards + leads to overfitting (undersmoothing)

Data was used twice → est. fcn → est. risk

tried to min $\frac{RSS}{n}$, so under est. risk

In Practice…

- Minimizing risk = balancing bias and variance



From Hastie, Tibshirani, Friedman

Approx 2: Cross Validation

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Mimic heldout data using *training* data
- Leave-one-out (LOO) cross validation (CV) algorithm:
  - Estimate fit using all but i^{th} data point $\hat{f}_{-i}^{\lambda}$ ← no obs. y_i
  - Predict i^{th} observation
  - Repeat for all i

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_{-i}^{\lambda}(x_i))^2$$

  - Repeat for all values of λ

Approx 2: Cross Validation

- Reasoning

$$CV = E\left[\left(Y_i - \hat{f}^{\lambda}_{-i}(x_i)\right)^2\right] = E\left[\left(Y_i - f(x_i) + f(x_i) - \hat{f}^{\lambda}_{-i}(x_i)\right)^2\right]$$

$$= \sigma^2 + E\left[\left(f(x_i) - \hat{f}^{\lambda}_{-i}(x_i)\right)^2\right]$$

$$\approx \sigma^2 + E\left[\left(f(x_i) - \hat{f}^{\lambda}_{n}(x_i)\right)^2\right]$$

- For linear smoothers

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}^{\lambda}_{n}(x_i)}{1 - L^{\lambda}_{ii}}\right)^2$$

only do fit once (per λ)

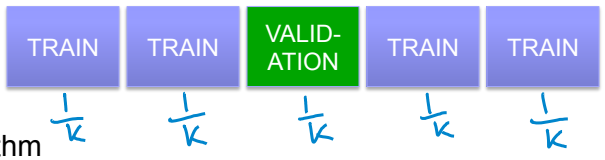
L^{λ}_{ii}: i^{th} diag. element of hat matrix using λ

- Warning: Curves can be very flat…Don't just choose and use without thinking. Some rules of thumb (see Elements of Statistical Learning)

Approx 2: Cross Validation

- K-fold cross validation

typically $K = 5, 10$



randomly assign obs. to each group

- Algorithm
  1. Fit model using data with k^{th} fraction removed
  2. Using fitted model, compute

$$\mathrm{CV}_k(\lambda) = \frac{1}{n_k}\sum_{i \in J(k)}\left(y_i - \hat{f}^{\lambda}_{-k}(x_i)\right)^2$$

w/o k^{th} block

$J(k)$: indices for k^{th} block

  3. Store

$$\mathrm{CV}(\lambda) = \frac{1}{K}\sum_{k=1}^{K}\mathrm{CV}_k(\lambda)$$

  4. Repeat for each value of λ using same split of the data

Approx 3: Generalized CV

- Recall LOO ordinary CV for linear smoothers

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - L_{ii}^{\lambda}}\right)^2$$

- Instead of L_{ii}^{λ}, use $\frac{1}{n}\sum_{i=1}^{n} L_{ii}^{\lambda} = \frac{1}{n}\mathrm{tr}(L^{\lambda}) = \frac{\nu_\lambda}{n}$

$$GCV(\lambda) = \frac{1}{n}\sum\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- Often very close to OCV solution

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2 \quad \mathrm{tr}(L^{\lambda})$$

- One motivation: Invariance to orthonormal transformations

$y, X \rightarrow Qy, QX$ where $Q^TQ = QQ^T = I$

$\min (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$ ⇕ $\min (Qy - QX\beta)^T(Qy - QX\beta) + \lambda\beta^T\beta$ — but not the same OCV score ($L_{ii}^{\lambda} = L_{Q,ii}^{\lambda}$)

If L^{λ} is a linear smoother for original data, $L_Q^{\lambda} = QL^{\lambda}Q^T$ is for trans. data

$\mathrm{tr}(L_Q^{\lambda}) = \mathrm{tr}(QL^{\lambda}Q^T) = \mathrm{tr}(L^{\lambda}\underbrace{Q^TQ}_{I}) = \mathrm{tr}(L^{\lambda})$

GCV scores will be the same

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

$(1-x)^{-2}$

- Using $(1-x)^{-2} \approx 1 + 2x$

$\hat{\sigma}^2$

$$GCV(\lambda) \approx \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_n^{\lambda}(x_i))^2 + \frac{2\nu_\lambda}{n}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_n^{\lambda}(x_i))^2\right)$$

$\approx$ Mallow's C_p stat

(but not exactly the right $\hat{\sigma}^2$...)

Approx 4: Mallows C_p Statistic

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

for linear smoothers

- **Solution:** Approximate directly

$$\text{avg. MSE} = \frac{1}{n}E\left[(\underline{f} - \underline{\hat{f}}_n^{\lambda})^T(\underline{f} - \underline{\hat{f}}_n^{\lambda})\right] = \frac{1}{n}E\left[(Y - \epsilon - L^{\lambda}Y)^T(Y - \epsilon - L^{\lambda}Y)\right]$$

f, $\hat{f}_n^{\lambda}$

$$= \frac{1}{n}E\left[(Y - L^{\lambda}Y)^T(Y - L^{\lambda}Y)\right] - \sigma^2 + \frac{2}{n}\nu_\lambda\sigma^2$$

uses $E[\epsilon^T L^{\lambda}\epsilon] = E[tr(\epsilon^T L^{\lambda}\epsilon)] = E[tr(L^{\lambda}\epsilon\epsilon^T)]$

$= tr(L^{\lambda} I \sigma^2) = \sigma^2\nu_\lambda$

Approx 4: Mallows C_p Statistic

$$\text{avg. MSE} = \frac{1}{n} E\left[(Y - L^\lambda Y)^T (Y - L^\lambda Y)\right] - \sigma^2 + \frac{2}{n}\nu_\lambda \sigma^2$$

- Estimate avg. MSE as

$$\frac{1}{n} RSS^\lambda - \frac{1}{n}(n - 2\nu_\lambda)\hat{\sigma}^2_{max}$$

using a maximal model

$$\Updownarrow$$

min Mallow's Cp

$$\frac{RSS^\lambda}{\hat{\sigma}^2_{max}} - (n - 2\nu_\lambda)$$

- Note: Arises from considering L_2 loss. Log-likelihood loss leads to AIC. For BIC, consider Bayesian model selection

Bayesian Model Selection

- Assume some *M* possible models
  - Model M_m *m=1,…,M* has parameters θ_m and prior $p(\theta_m \mid M_m)$
  - Prior over models $p(M_m)$
- Model posterior

training data

$$p(M_m \mid Z) \propto p(M_m) p(Z \mid M_m)$$

$$\propto p(M_m) \int p(Z \mid \theta_m, M_m) p(\theta_m \mid M_m) d\theta_m$$

eg mean + cov of Gauss or β_j

- Compare models:

$$\frac{p(M_m \mid Z)}{p(M_\ell \mid Z)} = \frac{p(M_m)}{p(M_\ell)} \frac{p(Z \mid M_m)}{p(Z \mid M_\ell)} \gtrless 1$$

posterior odds

often, uniform prior

Bayes factor

Bayesian Model Selection

- For Bayes factor, approximate (Laplace + ...; ML est.; # of free params)

$$\log p(Z \mid M_m) \approx \log p(Z \mid \hat{\theta}_m, M_m) - \frac{\nu_m}{2} \log n + O(1)$$

- If loss is $-2 \log p(Z \mid \hat{\theta}_m, M_m)$, then equivalent to BIC
  - Minimizing BIC = maximizing approximated posterior
- However, in addition to being able to select the best model, in Bayesian framework we also get the relative merit of each

$$\approx \frac{e^{-\frac{1}{2}\mathrm{BIC}_m}}{\sum_{\ell=1}^{M} e^{-\frac{1}{2}\mathrm{BIC}_\ell}}$$

- BIC is asymptotically consistent, but AIC is not
- For finite samples, BIC tends to choose too simple models

Reading

- Hastie, Tibshirani, Friedman: 7.2 (again), 7.4-7.7, 7.10
- Wakefield: 10.6 (up to 10.6.4)

What you should know…

- Model selection vs. model assessment tasks
- Training/validation/test split
- In-sample approaches for selecting the smoothing parameters:
  - Training error = BAD
  - Cross validation (CV)
    - LOO
    - K-fold
  - Generalized cross validation (GCV)
  - Mallow's C_p
- Bayesian model selection

Module 2: Splines and Kernel Methods

Spline Model Overview, Regression Splines, Smoothing Splines

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 8th, 2014

Moving Beyond Linearity

- So far we have assumed standard linear models

$$\min_{\beta} \|y - X\beta\|_2^2 \quad \leftarrow \quad f(x) = \beta^T x$$

- In the case of many predictors relative to number of observations, we considered penalized regression to avoid overfitting

$$\min_{\beta} \|y - X\beta\|_2^2 + \lambda \|\beta\|$$

- Often a convenient form, and necessary to assume simple structure to avoid overfitting in data-scarce regimes, but linear assumption rarely holds in practice

Moving Beyond Linearity

- Consider generic functional forms (univariate *x* for now)

$$\min_{f} \|y - f(x)\|_2^2$$

  - If constrained to linear forms → LS soln
  - If arbitrary → interpolator … overfitting
- As before, penalize complexity. Here, in terms of roughness.

$$\min_{f} \|y - f(x)\|_2^2 + \lambda \int f''(x)^2 dx$$

  - If $\lambda \to 0$, interpolator
  - If $\lambda \to \infty$, LS soln (line … no 2nd der.)
- Remarkable result: Explicit, finite-dimensional minimizer

TBD natural cubic spline w/ knots at data pts

"smoothing spline"

Backtrack a bit…

- Instead of just considering input variables *x* (potentially mult.), augment/replace with transformations = "input features"
- ***Linear basis expansions*** maintain linear form in terms of these transformations

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x)$$

trans. ← $h_m(x)$

linear in these transformations →

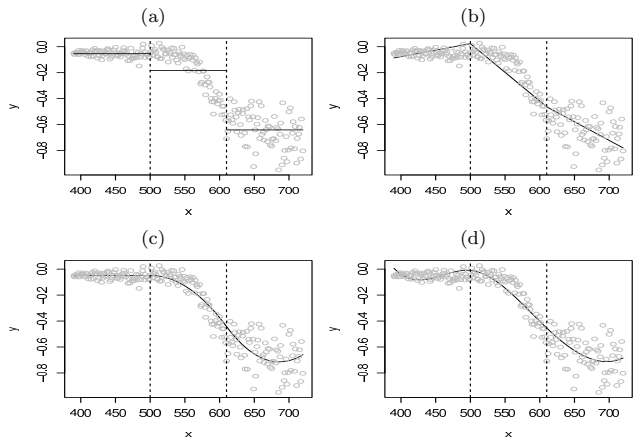
- What transformations should we use?
  - $h_m(x) = x_m$ → linear model
  - $h_m(x) = x_j^2, \quad h_m(x) = x_j x_k$ → polynomial reg.
  - $h_m(x) = I(L_m \leq x_k \leq U_m)$ → piecewise constant
  - …

Piecewise Polynomial Fits

- Again, assume *x* univariate
- Polynomial fits are often good locally, but not globally
  - Adjusting coefficients to fit one region can make the function go wild in other regions
- Consider ***piecewise polynomial*** fits
  - Local behavior can often be well approximated by low-order polynomials

Piecewise Polynomial Fits

LIDAR Data Example



From Wakefield book

Piecewise Constant/Linear Fits

- **Example 1: Piecewise constant, with 3 basis functions**

$h_1(x) = \mathbb{I}(x \leq \xi_1)$ "knot"

$h_2(x) = \mathbb{I}(\xi_1 \leq x \leq \xi_2)$

$h_3(x) = \mathbb{I}(\xi_2 \leq x)$

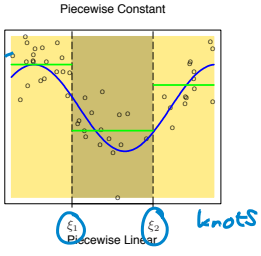
- Resulting model: $f(x) = \sum_{m=1}^{3} \beta_m h_m(x)$

- Fit: Take mean of data in each region

$\hat{\beta}_m = \bar{Y}_m \leftarrow$ data in region

- **Example 2: Piecewise linear**
- Add three basis functions:

$h_{m+3}(x) = h_m(x)x$ $m=1,2,3$



From Hastie, Tibshirani, Friedman book

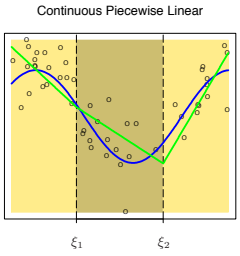
Regression Splines – Linear

- Resulting piecewise linear model:

$$f(x) = I(x < \xi_1)(\beta_1 + \beta_4 x) + I(\xi_1 \leq x < \xi_2)(\beta_2 + \beta_5 x) + I(\xi_2 \leq x)(\beta_3 + \beta_6 x)$$

  - # of params? 6

- Typically prefer continuity...
  - Enforce $f(\xi_1^-) = f(\xi_1^+)$ $f(\xi_2^-) = f(\xi_2^+)$
  - Which implies $\beta_1 + \beta_4 \xi_1 = \beta_2 + \beta_5 \xi_1$ $\beta_2 + \beta_5 \xi_2 = \beta_3 + \beta_6 \xi_2$
  - # params? $6 - 2 = 4$

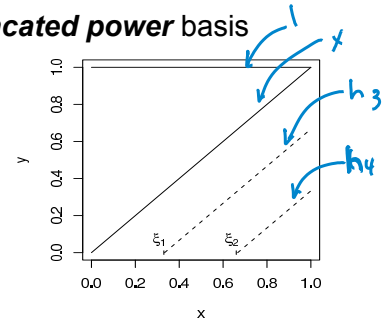


From Hastie, Tibshirani, Friedman book

Regression Splines – Linear

- More directly, we can use the ***truncated power*** basis

$$h_1(x) = 1$$
$$h_2(x) = x$$
$$h_3(x) = (x - \xi_1)_+$$
$$h_4(x) = (x - \xi_2)_+$$



From Wakefield book

- Resulting model: $f(x) = \beta_0 + \beta_1 x + \beta_2 (x - \xi_1)_+ + \beta_3 (x - \xi_2)_+$

- Continuous at the knots because all prior basis functions are contributing to the fit up to any single *x*

Reading

- Hastie, Tibshirani, Friedman: 5.1-5.5 (skipping 5.3)
- Wakefield: 11.1.1-11.2.3

What you should know…

- Linear basis expansions
- Regression splines
  - Cubic splines, natural cubic splines, …
  - Interpretation as a linear smoother
  - Degrees of freedom
- Smoothing splines
  - Arising from penalized regression setting with smoothness penalty
  - Cubic spline basis with knots at every data point